Using Artificial Intelligence Planning to Generate Antenna Tracking Plans

Forest Fisher, Tara Estlin, Darren Mutz, and Steve Chien

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
{firstname.lastname}@jpl.nasa.gov

Abstract

This paper describes the application of Artificial Intelligence planning techniques to the problem of antenna track plan generation for a NASA Deep Space Communications Station. The described system enables an antenna communications station to automatically respond to a set of tracking goals by correctly configuring the appropriate hardware and software to provide the requested communication services. To perform this task, the Automated Scheduling and Planning Environment (ASPEN) has been applied to automatically produce antenna tracking plans that are tailored to support a set of input goals. In this paper, we describe the antenna automation problem, the ASPEN planning and scheduling system, how ASPEN is used to generate antenna track plans, the results of several technology demonstrations, and future work utilizing dynamic planning technology.

INTRODUCTION

The Deep Space Network (DSN) [4] was established in 1958 and since then it has evolved into the largest and most sensitive scientific telecommunications and radio navigation network in the world. The purpose of the DSN is to support unmanned interplanetary spacecraft missions and support radio and radar astronomy observations in the exploration of the solar system and the universe. The DSN currently consists of three deep-space communications facilities placed approximately 120 degrees apart around the world: at Goldstone, in California's Mojave Desert; near Madrid, Spain; and near Canberra, Australia. This strategic placement permits constant observation of spacecraft as the Earth rotates, and helps to make the DSN the largest and most sensitive scientific telecommunications system in the world. Each DSN complex operates four deep space stations -- one 70-meter antenna, two 34-meter antennas, and one 26-meter antenna. The functions of the DSN are to receive telemetry signals from spacecraft, transmit commands that control the spacecraft operating modes, generate the radio navigation data used to locate and guide the spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry, and geodynamics measurements.

From its inception the DSN has been driven by the need to create increasingly more sensitive telecommunications devices and better techniques for navigation. The operation of the DSN communications complexes requires a high level of manual interaction with the devices in the communications link with the spacecraft. In more recent times NASA has added some new drivers to the development of the DSN: (1) reduce the cost of operating the DSN, (2) improve the operability, reliability, and maintainability of the DSN, and (3) prepare for a new era of space exploration with the New Millennium program: support small, intelligent spacecraft requiring very few mission operations personnel [8].

This paper addresses the problem of automated track plan generation for the DSN, i.e. automatically determining the necessary actions to set up a communications link between a deep space antenna and a spacecraft. Similar to many planning problems, track plan generation involves elements such as subgoaling to achieve preconditions and decomposing high-level (abstract) actions into more detailed sub-actions. However, unlike most classical planning problems, the problem of track generation is complicated by the need to reason about issues such as metric time, DSN resources and equipment states. To address this problem, we have applied the Automated Scheduling and Planning Environment (ASPEN) to generate antenna track plans on demand.

ASPEN [1,7] is a generic planning and scheduling system being developed at JPL that has been successfully applied to problems in both spacecraft commanding and maintenance scheduling and is now being adapted to generate antenna track plans. ASPEN utilizes techniques from Artificial Intelligence planning and scheduling to automatically generate the necessary antenna command sequence based on input goals. This sequence is produced by utilizing an "iterative repair" algorithm [7,9,12], which classifies conflicts and resolves them each individually by performing one or more plan modifications. This system has been adapted to input antenna tracking goals and automatically produce the required command sequence to set up the requested communications link.

This work is one element of a far-reaching effort to upgrade and automate DSN operations. The ASPEN Track Plan Generator has been demonstrated in support of the Deep Space Terminal (DS-T), which is a prototype 34-meter deep space communications station intended to be capable of fully autonomous operations [5,6].

This rest of this paper is organized in the following manner. We begin by characterizing the current mode of operations for the DSN, and then describe the track plan generation problem. Next, we introduce the ASPEN planning and scheduling system and describe its modeling language and search algorithm(s). We then present an operations example of using this system for track plan generation and discuss several successful demonstrations that were performed with Mars Global Surveyor using a 34-meter antenna station in Goldstone, CA. Finally, we discuss some related work and describe current efforts to expand this system to incorporate a dynamic planning approach which will allow for closed-loop control and automatic error recovery when executing a DSN antenna track.

HOW THE DSN OPERATES

The DSN track process occurs daily for dozens of different NASA spacecraft and projects which use the DSN to capture spacecraft data. Though the process of sending signals from a spacecraft to Earth is conceptually simple, in reality there are many earthside challenges that must be addressed before a spacecraft's signal is acquired and successfully transformed into useful information. In the remainder of this section, we outline some of the steps involved in providing tracking services and in particular discuss the problem of track plan generation.

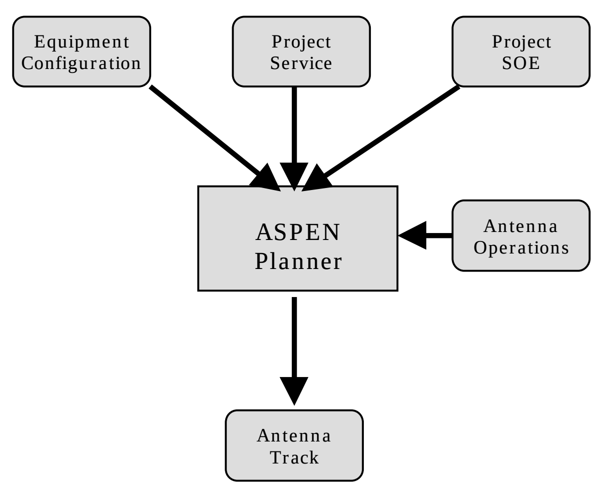
The first step in performing a DSN track is called network preparation. Here, a project sends a request for the DSN to track a spacecraft involving specific tracking services (e.g. downlink, uplink). The DSN responds to the request by attempting to schedule the necessary resources (i.e. an antenna and other shared equipment) needed for the track. Once an equipment schedule and other necessary information has been determined, the next step is the data capture process, which is performed by operations personnel at the deep space station. During this process, operators determine the correct steps to perform the following tasks: configure the equipment for the track, perform the actual establishment of the communications link, and then perform the actual track by issuing control commands to the various subsystems comprising the link.

Throughout the track the operators continually monitor the status of the link and handle exceptions (e.g. the receiver breaks lock with the spacecraft) as they occur. All of these actions are currently performed by human operators, who manually issue tens or hundreds of commands via a computer keyboard to the link subsystems. This paper discusses the application of the ASPEN planning system to automatically generate DSN track plans (i.e. the steps necessary to set up and perform the requested track) and dramatically reduce the need for many manual steps.

TRACK PLAN GENERATION: THE PROBLEM

Generating an antenna track plan involves taking a general service request (such as telemetry - the downlink of data from a spacecraft), an antenna knowledge-base (which provides the information on the requirements of antenna operation actions), and other project specific information (such as the spacecraft sequence of events), and then generating a partially-ordered sequence of commands. This command sequence will properly configure a communications link that enables the appropriate interaction with the spacecraft. To automate this task, the ASPEN planning and scheduling system has been applied to generate antenna operation procedures on demand.

ASPEN has been adapted to use high-level antenna track information to determine the appropriate steps, parameters on these steps and ordering constraints on these steps that will achieve the input track goals. In generating the antenna track plan, the planner uses information from several sources (see Figure 1):



```
1     Activity Pre_track {
2        Decompositions =
3            (Begin_pre_track, Configure_subsystems, Point_antenna, On_point_check,
4             Start_APC where ordered)
5     };
6
7      Activity Acquire_signal {
8        int way;
9        time_param bot_time;
10       Timeline_dependencies =
11           bot_time <- bot_time_sv, way <- way_sv;
12       reservations =
13           BVR,
14           Antenna_sv must_be “on_point”,
15           Signal_sv change_to “acquired”;
16    };
```

Figure 2 ASPEN Activity Examples

Figure 1 ASPEN Inputs and Outputs

Project Service Request **-** The service request specifies the DSN services (e.g. downlink, uplink) requested by the project and corresponds to the goals or purpose of the track.

Project SOE **-** The project sequence of events (SOE) details spacecraft events occurring during the track - including the timing of the beginning and ending of the track and spacecraft data transmission bit rate changes, modulation index changes, and carrier and subcarrier frequency changes.

Antenna Operations KB **-** The Antenna Operations Knowledge Base (KB) stores information on available antenna operations actions/commands. This KB dictates how actions can be combined to provide essential communication services. Specifically, this includes information such as action preconditions, postconditions, and command directives and also includes any other relevant information such as resource and state descriptions.

Equipment Configuration **-** This configuration details the types of equipment available and includes items such as the antenna, antenna controller, the receiver, etc.

THE ASPEN MODELING LANGUAGE AND SEARCH ALGORITHM

ASPEN is a reusable, configurable, generic planning/ scheduling application framework that can be tailored to specific domains to create conflict-free plans or schedules. Its components include:

- An expressive modeling language to allow the user to naturally define the application domain
- A constraint management system for representing and maintaining antenna and/or spacecraft operability and resource constraints, as well as activity requirements
- A set of search strategies
- A temporal reasoning system for expressing and maintaining temporal constraints
- A graphical interface for visualizing plans/schedules

A brief introduction into the ASPEN modeling language is given below. For more details on ASPEN, see [1,7].

Modeling Language

The ASPEN modeling language allows the user to define activities, resources and states that describe a particular application domain. A domain model is input at run-time, so modifications can be made to the model without requiring ASPEN to be recompiled. The modeling language has a simple syntax, which can easily be used by operations personnel. Each application model is comprised of several files, which define and instantiate activities, resources and states.

The central data structure in ASPEN is an activity. An activity corresponds to the act of performing a certain function (e.g. configuring the antenna receiver) and represents an action or step in a plan/schedule. Once instantiated it has a start time, an end time, and duration. Activities can also use one or more resources and reason about domain states. Figure 2 shows several activity definitions utilized for antenna-track plan generation. Shown is a “Pre_track” activity that introduces into the plan the steps required to set up the antenna and subsystems for the actual track, and an “Acquire_signal” activity that uses the antenna receiver to acquire the spacecraft signal.

Activity parameters are used to store values in activities or reservations. Lines 8 and 9 contain parameters that specify the number of communication channels (or ways) utilized in the track and the time the track began. Parameter values can be set in an activity definition, passed in from other activities, or as in this case, determined by checking the value of a particular state (as shown on lines 10 and 11). These parameter values are then later referenced when generating the actual command that will execute this step in the final antenna track plan.

Activities can also contain decompositions, as shown in the first activity definition in Figure 2. This activity contains a decomposition into several subactivities (e.g. Configure_subsystems, Point_antenna). These subactivities are activities that can be scheduled any time within the parent activity time interval subject to any constraints within the subactivity definitions. Thus as soon as a "Pre_track" activity is instantiated in a plan, it's subactivities are also instantiated. Decompositions may also be "ordered", such as the one shown here, where all sub-activities must occur in the order specified.

Reservations are used to reserve a portion of a resource or state for the duration of an activity. The second activity in Figure 2 contains a reservation on the Block-V Receiver (BVR). There are two main types of resource reservations in ASPEN: atomic and aggregate. Line 13 of Figure 2 shows an example of an atomic reservation that reserves the BVR for the duration of the activity. No other activities can use the BVR during this time. An example of an aggregate reservation would be to use N units of power or fuel or some other depletable resource.

State reservations can be used to require a certain state be true or change the value of a state variable. Line 14 of Figure 2 requires that the antenna be "on_point" (indicating that the antenna is pointing at the correct set of coordinates) before attempting to acquire the spacecraft signal. Line 15 changes the state of the signal state variable to "acquired" indicating that the spacecraft signal has been successfully acquired by the receiver.

One other utilized feature that is not shown is temporal constraints between activities. Examples of these constraints are: starts_before, starts_after, contains, etc. These constraints can be used to specify partial ordering over certain activities. For example, in the antenna track generation model, it's specified that the activity for generating receiver predicts (where predicts dictate settings for the receiver) must be ordered before the activity which delivers the predicts to the receiver (e.g. Generate_ bvr_predicts ends_before start of Deliver_bvr_predicts).

Besides activities, other defined model elements include resources and states. Resource definitions contain a profile of a physical resource over time. There are three main types of resources: atomic, depletable, and non-depletable. Atomic resources are physical devices that can only be used (reserved) by one activity at a time, such as a receiver or antenna controller. Depletable resources are resources that can be used by more than one activity at a time, but their capability is diminished after use, such as a battery or other power source. Non-depletable resources are similar to depletable resources except that their capacity does not diminish and thus they do not need to be replenished, such as memory bus. Most of the resources utilized for antenna track plan generation are atomic resources that represent different pieces of equipment.

A device or subsystem may also be represented by a state variable that gives information about its state over time. A state variable contains a state profile, which is defined as an enumerated type. Some examples of possible states are that an antenna can be "on_point", "off_point" or "stowed", a receiver can be "locked" or "unlocked" and the Conscan subsystem can be "on" or "off." States can be reserved or changed by activities and a state variable must equal some state at every time. Also, if there are several different states possible for a particular state variable, allowable state transitions can be defined where only certain transitions between those states are possible.

Conflict Detection

Conflicts arise within a plan when a constraint has been violated. This constraint could be temporal or involve a parameter, resource or state. In order to reason about temporal constraints, ASPEN utilizes a Temporal Constraint Network (TCN) that describes temporal relationships between activities. The TCN can be queried as to whether the temporal constraints currently imposed between activities are consistent. Also used is a Parameter Dependency Network (PDN) that reflects any defined dependencies between activity parameters. A dependency between two parameters is defined as a function from one parameter to another. These dependencies are maintained by the PDN which checks that at any given time all dependency relations are satisfied.

Resource timelines are used to reason about the usage of physical resources by activities. Conflicts are detected if two or more activities are utilizing an atomic resource at the same time or if the aggregate usage of a resource exceeds its capacity at any given time. State timelines represent attributes, or states, that can change over time where each state can have several possible values. As activities are placed/moved in time, the state timeline updates the values of the state, and detects possible inconsistencies or conflicts that can be introduced as a result.

Planning/Scheduling Algorithms

The search algorithms utilized in a planning/scheduling system typically search for a valid, possibly near-optimal plan/schedule. The ASPEN framework has the flexibility to support a wide-range of scheduling algorithms. For

```
Pre_track pre_track1{
  Start_time = 1998-213/13:32:26;
  End_time = 1998-213/13:47:26;
};

Track Track1{
  Start_time = 1998-213/13:47:26;
  End_time = 1998-213/16:40:00;
};

Post_track post_track1{
  Start_time = 1998-213/16:40:00;
  End_time = 1998-213/16:50:00;
};
```

Figure 3 Activity Instantiations

this application, we mainly utilized a repair-based algorithm [7,9,12].

An iterative repair algorithm classifies conflicts and attacks them each individually. Conflicts occur when a plan constraint has been violated, where this constraint could be temporal or involve a resource, state or activity parameter. Conflicts are resolved by performing one or more schedule modifications such as moving, adding, or deleting activities. The iterative repair algorithm continues until no conflicts remain in the schedule, or a timeout has expired.

For track plan generation, ASPEN begins by generating a possibly invalid complete schedule using a greedy, constructive algorithm. Then, at every iteration, the schedule is analyzed, and repair heuristics that attempt to eliminate conflicts in the schedule are applied until a valid schedule is found. Domain-dependent heuristics can also be added to direct the search towards more optimal solutions.

TRACK PLAN GENERATION: AN EXAMPLE

Given a set of tracking requests, ASPEN can generate a conflict-free track plan within the order of seconds that will correctly set up the requested communications link. In order to begin the planning process, the tracking service request, the equipment configuration, and the project SOE are parsed and relevant information is placed in a initial setup file which lists the requested track goals and any relevant initial state information. For example, Figure 3 shows three activity instantiations that request that a “Pre_track”, “Track” and “Post_track” activity be placed in the final plan at specific times.

ASPEN then decomposes these activities into the necessary steps that set up the antenna and subsystems (i.e. “Pre_track”), that perform the track (i.e. “Track”), and that perform the necessary shutdown procedures once

```
Configure_equipment:

Start jsc_asn.prc(dss,sc,pass,&ret_status)
If (!ret_status) then
    Write(“fatal error: cannot start
pass”)
    Goto fatal_err
Endif

Start ugc_hi.prc
If (!ret_status) then
    Write(“fatal error: can’t control
UGC”)
    Goto fatal_err
Endif

Start apc_hi.prc
If (!ret_status) then
    Write(“fatal error: can’t control
APC”)
    Goto fatal_err
Endif
.
.
.

Point_antenna:

Ret_status = exec(“APC DCOS”)
Start apc_track.prc(&ret_status)
If (!ret_status) then
    Write(“fatal error: cannot point ant”)
    Goto fatal_err
Endif
```

the track had ended (i.e. “Post_track”). Other initial state information is provided in a “Set_state_values” activity, which sets up the appropriate state variables. The information includes the spacecraft ID, antenna ID, the tracking goals, the carrier and sub-carrier frequency, the symbol rate, etc. ASPEN is also provided with the model files that hold the relevant activity, parameter, resource and state definitions, which were explained in the previous section.

Once the initial goals and state information are loaded, ASPEN utilizes its iterative repair algorithm to create a conflict-free track plan that provides the requested services. This final plan contains a large amount of information, including a list of grounded activities (where each activity has been assigned a start time and end time), and a list of constraints over those activities, including temporal, parameter, resource and state constraints. ASPEN also displays the final resource and state timelines which show the states of those entities over the course of the plan. The actual antenna control script that will be used to execute the track is output in a separate file which contains the command sequence necessary to set up, control and break down the link. In the model definition, a command (or set of commands) can be specified for each defined activity. These commands are then output in the correct sequence based on the final plan constraints. An example of this file format is shown in Figure 4. This control script is then sent to an antenna operator or execution agent where it will be used to perform the requested track.

DS-T DEMONSTRATIONS

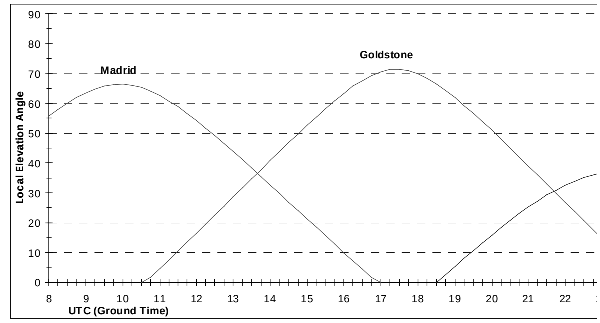
The Deep Space Terminal (DS-T) [5,6] being developed at the NASA Jet Propulsion Laboratory is a prototype 34-meter deep space communications station intended to be

Figure 5 34m BWG Antennas at Goldstone

capable of fully autonomous operations. When requested to perform a track, the DS-T station automatically performs a number of tasks (at appropriate times) required to execute the track. First, the Schedule Executive sets up the track schedule for execution and provides the means for automated rescheduling and/or manual schedule editing in the event of changes. The Configuration Engine is then responsible for retrieving all the necessary data needed for station operations. Next, the Script Generator (ASPEN) generates the necessary command sequence to perform the track. Finally, a Station Monitor and Control process executes the generated script and records relevant monitor data generated during the track.

The DS-T concept was validated through a number of demonstrations. The demonstrations began with the automation of partial tracks in April 1998, continued with 1-day unattended operations in May, and concluded with a 6-day autonomous "*lights-out*" demonstration in September 1998. Throughout these demonstrations ASPEN was used to automatically generate the necessary command sequences for a series of Mars Global Surveyor (MGS) downlink tracks using the equipment configuration at Deep Space Station 26 (DSS26), a 34-meter antenna located in Goldstone, CA. These command sequences were produced and executed in a fully autonomous fashion with no human intervention. During the September demonstration performed all Mars Global Surveyor coverage scheduled for the Goldstone antenna complex. This corresponded to roughly 13 hours of continuous track coverage per day.

In Figure 5, we show a picture of the three 34-meter Beam Wave Guide (BWG) antennas at the Goldstone, CA



Track Segments:

#1	11:15:00 – 14:00:23	3way/65	(Madrid)
#2	14:00:23 – 15:42:26	LOS	
#3	15:42:26 – 15:52:23	1way	
#4	15:42:26 – 15:52:23	3way/25	(Goldstone)
#5	15:52:23 – 20:03:34	1way	
#6	20:03:34 – 23:40:00	3way/34	(Canberra)

Figure 6 September 16, 1998 MGS Track

facility. In the foreground is DSS-26, which was the station selected for prototyping the DS-T.

While the overall DS-T effort consisted of a large team and a project duration of approximately 1.5 years, the DS-T automation team consisted of three team members. Of this teams work, approximately one work year was spent on the script generation effort. This effort primarily consisted of knowledge acquisition and model development, while a small effort was made in the integration of the script generator. A key factor in the quick development was the ability to adapt a general purpose planning and scheduling system. As the domain of ground communication-station commanding shared many similarities to spacecraft commanding, ASPEN seemed like a logical choice. This was confirmed by the ease of knowledge base development and integration. Spacecraft commanding also consists of generating a sequence of commands, however it is predominately a resource-scheduling problem, whereas ground-station commanding is predominately a sequencing problem.

RESULTS

In order to provide qualitative results, we present statistical data from September 16, 1998, a representative day during our 6-day autonomous unattended demonstration, durring which we collected above 90% of the transmitted frames. This performance is on par with the operator-controlled stations, however required no support personnel (i.e. reduced operations cost).

In figure 6, the graph represents when MGS was in view of the ground stations at each of the three complexes (Madrid, Goldstone, and Canberra). DS-T, which is located at Goldstone, tracked MGS through the five track segments indicated in the figure 6.

Before continuing with the analysis of the results, let us explain the different modes indicated in figure 6 for each of the different track segments. When a spacecraft is downlinking data it is said to be in 1way mode. When an uplink and a downlink are taking place simultaneously the spacecraft is said to be in 2way mode. If a station is communicating in 2way mode with a spacecraft, and another station is listening in on the downlink of the spacecraft, the second station is said to be in 3way with the 2way station. Because DS-T is not equipped for uplink , DS-T operates in either 1way or 3way mode. Because the downlink frequency is relative to the uplink frequency, it is critical to determine the station involved in the uplink when taking part in a 3way mode of operations. In this example, during segment 4 dss25 (deep space station) was in 2way and DS-T was in 3way with 25 (3way/25).

Track segment 2, which is labeled LOS, indicates that there was a scheduled loss of signal (LOS) so during this segment no frames were collected. During each of the other respective track segment DS-T collected 75%, 91%, 96%, 90%, 23% of the broadcasted frames. As shown by the graph, during segment 1 and 6 the elevation of the dish is low in the sky. Under these circumstances there is considerably more atmospheric interference which explains the lower percent of frame collection. On the other hand, if you look at segment 4 where there is a long segment with the spacecraft high in the sky the data collection is quite high. In segment 3 and 5 the values are a little lower due to the shortness of the segments. This is explained by the fact that some data is lost during a change in mode, as in the transition from LOS to 1way and 3way/25 to 1way.

As a component of the DS-T demonstrations, the SG performed flawlessly, producing dynamically instantiated control scripts based on the desired service goals for the communications pass as specified in the service request. The use of such technology resulted in a three primary benefits:

- Autonomous operations enabled by eliminating the need for hundreds of manual inputs in the form of control directives. Currently the task of creating the communications link is a manual and time-consuming process which requires operator input of approximately 700 control directives and the constant monitoring of several dozen displays to determine the exact execution status of the system.
- Reduced the level of expertise of an operator required to perform a communication track. Currently the complex process requires a high level of expertise from the operator, but through the development of the KB by a domain expert this expertise is captured with in the system itself.
- The KB provides a declarative representation of operation procedures. Through the capture of this expertise the KB documents the procedural steps of performing antenna communication services.

RELATED WORK

There are a number of existing systems built to solve real-world planning or scheduling problems [10,11,12]. The problem of track plan generation combines elements from both these fields and thus traditional planners and schedulers cannot be directly applied. First, many classical planning elements must be addressed in this application such as subgoaling to achieve activity preconditions (e.g. the antenna must be "on_point" to lock up the receiver) and decomposing higher-level (abstract) activities into more detailed sub-activities. In addition, many scheduling elements are presents such as handling metric time and temporal constraints, and representing and reasoning about resources (e.g. receiver, antenna controller) and states (e.g. antenna position, subcarrier frequency, etc.) over time.

One other system has been designed to generate antenna track plans, the Deep Space Network Antenna Operations Planner (DPLAN) [2]. DPLAN utilizes a combination of AI hierarchical-task network (HTN) and operator-based planning techniques. Unlike DPLAN, ASPEN has a temporal reasoning system for expressing and maintaining temporal constraints and also has the capability for representing and reasoning about different types of resources and states. ASPEN can utilize different search algorithms such as constructive and repair-based algorithms, where DPLAN uses a standard best-first based search. And, as described in the next section, ASPEN is currently being extended to perform dynamic planning for closed-loop error recovery, where DPLAN has only limited replanning capabilities.

FUTURE WORK: PROVIDING CLOSED-LOOP CONTROL THROUGH DYNAMIC PLANNING

Currently, we are working on modifying and extending the current ASPEN Track Plan Generator to provide a Closed Loop Error Recovery system (CLEaR) for DSN track automation. CLEaR is a real-time planning system built as an extension to ASPEN [3]. The approach taken is to dynamically feed monitor data (sensor updates) back into the planning system as state updates. As these dynamic updates come in, the planning system verifies the validity of the current plan. If a violation is found in the plan, the system will perform local modification to construct a new valid plan. Through this continual planning approach, the plan is disrupted as little as possible and the system is much more responsive and reactive to changes in the real (dynamic) world.

This CLEaR effort is also being integrated with a Fault Detection, Isolation and Recovery (FDIR) system. FDIR is an expert system providing monitor data analysis. As is often the case with large complex systems, monitor (sensor) data is often related in different ways that becomes difficult for a human to detect. The advantage of combining these two systems is that FDIR can first interpret the vast amount of data and summarize it into a set of meaningful values for a planning system to react to. We think of this union as intelligent analysis and intelligent response, much like a careful design and implementation; one without the other is of little use.

CONCLUSIONS

This paper has described an application of the ASPEN automated planning system for antenna track plan generation. ASPEN utilizes a knowledge base of information on tracking activity requirements and a combination of Artificial Intelligence planning and scheduling techniques to generate antenna track plans that will correctly setup a communications link with spacecraft. We also described several demonstrations that have been performed as part of the DS-T architecture where ASPEN was used to generate plans for downlink tracks with Mars Global Surveyor. Finally, we described a planned extension of this system, which will allow for closed-loop control, error recovery and fault detection using dynamic planning techniques.

ACKNOWLEDGEMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract of the National Aeronautics and Space Administration. We thank members of the ASPEN scheduling team and members of the DS-T automation team for contributing to this work.

References

[1] S. Chien, D. Decoste, R. Doyle, and P. Stolorz, “Making an Impact: Artificial Intelligence at the Jet Propulsion Laboratory,” AI Magazine, 18(1), 103-122, 1997.

[2] S. Chien, R. Hill Jr., A. Govindjee, X. Wang, T. Estlin, A. Griesel, R. Lam and K. Fayyad, “A Hierarchical Architecture for Resource Allocation, Plan Execution, and Revision for Operation of a Network of Communication Antennas,” Proceedings of the 1997 IEEE Conference on Robotics and Automation, Albuquerque, NM, April 1997.

[3] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, “Integrated Planning and Execution for Autonomous Spacecraft,” To appear in the Proceedings of the 1999 IEEE Aerospace Conference, Aspen, CO, March, 1999.

[4] Deep Space Network, Jet Propulsion Laboratory Publication 400-517, April 1994.

[5] F. Fisher, S. Chien, L. Paal, E. Law, N. Golshan, and M. Stockett, “An Automated Deep Space Communications Station,” Proceedings of the 1998 IEEE Aerospace Conference, Aspen, CO, March 1998.

[6] F. Fisher, D. Mutz, T. Estlin, L. Paal, and S. Chien, “The Past, Present and Future of Ground Station Automation with in the DSN,” To appear in the Proceedings of the 1999 IEEE Aerospace Conference, Aspen, CO, March 1999.

[7] A. Fukanaga, G. Rabideau, S. Chien, and D. Yan, “Toward an Application Framework for Automated Planning and Scheduling,” Proceedings of the 1997 International Symposium of Artificial Intelligence, Robotics and Automation for Space, Tokyo, Japan, July 1997.

[8] R. W. Hill, Jr., S. A. Chien, K. V. Fayyad, C. Smyth, T. Santos, and R. Bevan, “Sequence of Events Driven Automation of the Deep Space Network,” Telecommunications and Data Acquisition 42-124, October-December 1995.

[9] S. Minton and M. Johnston, “Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems,” Artificial Intelligence, 58:161-205, 1988.

[10] A. Tate, B. Drabble and R Kirby, “O-Plan2: An Open Architecture for Command Planning and Control,” Intelligent Scheduling (Eds. M. Fox & M. Zweben), Morgan Kaufmann, 1994.

[11] D. Wilkins Practical Planning: Extending the Classical AI Planning Paradigm, Morgan Kaufmann, 1994.

[12] M. Zweben, B. Daun, E. Davis, and M. Deale, Scheduling and Rescheduling with Iterative Repair, in Intelligent Scheduling, Morgan Kaufmann, 1994.